## REMARKS

### Status of Claims

Claims 1-33 are currently pending with claims 1, 6, 9, 14, 19, 24, and 29 being independent. No claims have been amended, canceled, or added by way of this Amendment.

### Office Action Summary

In the June 1, 2006, Office action, the Examiner rejected claims 1, 6, 9, 14, 24, and 29 under 35 U.S.C. § 101 and rejected all pending claims under 35 U.S.C. § 103(a) as being unpatentable over Xu (U.S. Patent No. 5,995,958) in view of Laitinen (U.S. Patent No. 5,862,383). Applicants respectfully submit that the claimed invention is patentable subject matter under § 101 and that the Examiner's combination of references does not disclose or suggest all claimed features of the present inventions.

### Rejections under 35 U.S.C. § 101

The Examiner rejected claims 1, 6, 9, 14, 24, and 29, for being non-statutory subject matter as they allegedly do not produce a useful, tangible, and concrete result (Office action, page 2). The Examiner's current § 101 rejections are nearly identical to the § 101 rejections she presented in the August 10, 2005, Office action. In a response filed August 30, 2005, Applicant explained at length how the claimed invention produces a useful, tangible, and concrete result by converting query language statements into an imperative language statement for execution by a computing device.

Execution of imperative language statements (such as C++ code) formed from query language statements (such as SQL queries) allows computing devices that are unable to execute query language statements (such as PDAs) to nevertheless manage or interact with databases. The Examiner apparently agreed that the claimed invention constitutes patentable subject matter as the previous § 101 rejections were withdrawn in the December 21, 2005, Office action. As the claimed invention produces a useful, concrete, and tangible result, Applicants respectfully submit that all pending claims constitute patentable subject matter under § 101.

## Rejections under 35 U.S.C. § 103(a)

The Examiner generally contends that the combination of Xu and Laitinen discloses various features claimed by the present application, such as receiving a query, interpreting the query by associating a declarative language function with the query, converting the interpreted query to an imperative language statement, and executing the imperative language statement (Office Action, page 3). Applicants respectfully submit that the Examiner's combination discloses none of these features, as Xu and Laitinen are—at best—only marginally related to the field of the claimed invention.

### *The Examiner's combination does not disclose or suggest receiving queries or interpreting queries*

The Examiner's § 103 rejections are founded on the premise that Xu discloses (1) receiving queries in a query language, and (2) interpreting the queries by associating at least one declarative language function with the terms that comprise the queries (Office action, page 3). Xu does not disclose or suggest either of these features as Xu merely provides a new database query language based on lambda calculus and does not "receive" queries in a query language and interpret the received queries.

Xu's lambda-calculus query language allows database management systems to have full computing capabilities without the need for application-dependent software development in a host programming language (Abstract). Thus, by using only the lambda-calculus query language, the database disclosed by Xu can perform high-level functions unable to be performed by conventional query languages like SQL (col. 6, ll. 5-23). As discussed in more detail below, Xu does not disclose receiving conventional queries (such as SQL queries) and converting them to the lambda-calculus language, as Xu requires the queries to be initially written in the lambda-calculus language (col. 13, ll. 11-25).

The Examiner cites Xu, col. 2, as disclosing the claimed feature of receiving a query in a query language. However, Xu, cols. 1-2, is merely discussing the operation of a conventional relational database system (col. 1, ll. 57-58) and the disadvantages associated therewith (col. 2, ll. 50-51). Xu does not disclose receiving queries because it is directed at a new type of query language—which is not intended to be converted into other languages (col. 14, ll. 11-26). Xu has no use for conventional queries, or the reception thereof, due to its complete reliance on the lambda-calculus based queries.

Further, Xu does not disclose "interpreting" queries in a query language by associating declarative language functions therewith. The Examiner cites Xu, cols. 13-14, as disclosing the interpretation feature claimed by the present application. However, Xu does not disclose or suggest interpreting received queries as Xu only discloses allowing a programmer to write and form queries using a calculus-based query language (col. 13, ll. 11-25; col. 13, ll. 42-45; col. 14, ll. 12-13; col. 14, ll. 22-23; etc). Thus, Xu, cols. 13-14, are directed at how queries are written in a calculus-based query language and not at associating declarative language functions (such as those found in LISP) with query terms in a query language (SQL for example).

Xu has no need for forming declarative language functions and imperative language statements because its calculus-based query language is intended to be the "only" language required to execute high-level functions (col. 6, ll. 5-7). In contrast, embodiments of the present invention provide a method for allowing limited-power computing devices (such as PDAs) to manage or interact with a database even though they are unable to execute query language statements (which would likely include Xu's calculus-based query language).

Laitinen, alone or in combination with Xu, also does not disclose or suggest this feature as Laitinen's teachings have absolutely no relevance to databases or database queries, as is discussed in more detail below.

*The Examiner's combination does not disclose converting queries represented by declarative language statements*

The Examiner concedes that Xu does not disclose or suggest various features claimed by the present application, such as the "converting the queries represented by the at least one declarative language function to a plurality of imperative language statements" feature recited in claim 1. The Examiner relies on the teachings of Laitinen in combination with Xu to reject all claims under § 103.

However, Laitinen does not disclose this, or any other, claimed feature. Laitinen is directed at a method for generating code for a digital signal processor and its teachings have no relevance to databases or database queries (Abstract; col. 1, ll. 5-10).

The portions of Laitinen cited by the Examiner discuss the conversion of technical specifications of a programming language between imperative and declarative forms of the programming language (col. 3, ll. 18-26). The "technical specifications" which are converted are not queries and do not relate to query languages and are instead "communication protocol specifications which control the encoding and decoding functions" of the digital signal processor (col. 3, ll. 26-30). Thus, Laitinen discloses converting communication protocol specifications between various levels of the programming language (TSN) to enable varying control of the digital signal processor. Laitinen does not disclose or suggest converting queries represented by declarative language statements—even when indiscriminately combined with Xu.

The dependent claims recite further features neither taught nor suggested by the Examiner's combination. For example, claims 2, 15, and 25 recite "converting the query language to an intermediate tree representation corresponding to the at least one declarative language function associated with the plurality of query terms, and thereafter converting the query to at least one data structure that is interpreted by an imperative language interpreter core to perform the queries," which is neither taught nor suggested by any reference of record. The Examiner fails to identify any portions of Xu or Laitinen that disclose or suggest this feature.

*Laitinen is non-analogous art*

Laitinen is non-analogous art to the claimed invention as it is not in the same field of endeavor as the claimed invention and it is not reasonably pertinent to the particular problem with which the claimed invention is involved. *See In re Clay,* 23 USPQ.2d 1058, 1060 (Fed. Cir. 1992). As discussed above, embodiments of the claimed invention are directed at methods for managing and interacting with databases and address the problems associated with computing devices that are unable to execute query languages such as SQL.

In contrast, Laitinen has nothing to with databases or database queries and is instead entirely directed at a method and system for generating code for operating a digital signal processor (Abstract; col. 1, ll. 5-10). As Laitinen is clearly non-analogous art, it cannot be used by the Examiner to form a *prima facie* case of obviousness.

*There is no suggestion or motivation to combine Xu and Laitinen*

The Examiner contends that the suggestion to combine Xu, a new database query language, and Laitinen, a code generator for digital signal processors, is that "it would enable the system to provide conversion of the technical specifications to the programming languages that [are] involved in the reprogramming" (Office action, page 4). Based on the disclosures of Xu and Laitinen, Applicants respectfully submit that the Examiner's provided motivation does not make any sense at it appears to be only a description of Laitinen and not an explanation as to why one would combine a database query language (Xu) with a code generator for a digital signal processor that is entirely unrelated to relational databases (Laitinen).

In addition to Laitinen's non-relevance to the present inventions, there can be no suggestion or motivation to combine Xu with other references for the purpose of converting Xu's calculus-based query language as Xu expressly discloses that the calculus-based query language's purpose is to eliminate the need for other languages (such as the claimed imperative languages).

Thus, the Examiner's proposed combination expressly teaches away from Xu and there can be no suggestion or motivation to form the combination.

## Conclusion

The claims of the present application are patentable subject matter as they provide a useful, tangible, and concrete result. The Examiner's cited references fail to disclose or suggest all claimed features of the present application as Xu and Laitinen, in any combination, do not disclose (1) receiving queries in a query language; (2) interpreting the queries by associating declarative language functions with the queries; or (3) converting the queries represented by declarative language functions to an imperative language. Accordingly, Applicants respectfully request that a timely Notice of Allowance be issued in this case.

Should the Examiner have any questions, please contact the undersigned at (800) 445-3460. While the undersigned does not believe any additional fees are due in connection with this application, the Commissioner is hereby authorized to charge any additional fees associated with this communication or credit any overpayment to Deposit Account No. 09-0460.

Respectfully submitted,

HOVEY WILLIAMS LLP

By:

Scott R. Brown, Reg No. 40,535
HOVEY WILLIAMS LLP
2405 Grand Boulevard, Suite 400
Kansas City, Missouri 64108
(816) 474-9050

ATTORNEYS FOR APPLICANTS